



Boolean

Transcripción

[00:00] Siguiendo aquí con el ejemplo que hemos dado, vamos a hacer un poco más legible este código, un poco más entendible. Si se dan cuenta, está así agrupado de acuerdo al scope de cada statement, pero aún podemos utilizarlo más. ¿Qué vamos a hacer? Vamos a crear otra clase, solamente para tener el antes y el después de cómo estamos manejando el proyecto.

[00:32] Entonces le damos a EjemploCondicionales2 y le damos a finish. Tenemos aquí EjemploCondicionales2, y vamos a hacer algo que normalmente ustedes aprenderían por su propia cuenta que es copiar y pegar el código, una práctica muy usada entre programadores. Venimos aquí, pegamos y listo, ya estamos aquí.

[01:06] Ahora, así como si yo daba doble clic aquí en la consola podía maximizar esta view, si le doy doble clic aquí ya tengo el espacio completo para trabajar aquí. Entonces, vamos a detectar algunos anti-patrones en este código. Por anti-patrón me estoy refiriendo a una mala práctica o algún error digamos en la declaración de alguna variable, alguna cosa que pueda ser mejorada.

[01:38] Por ejemplo aquí, cantidad. Yo no estoy especificando cantidad de qué, y ya habíamos dicho antes que en Java solemos declarar explícitamente, nombrar explícitamente a qué está haciendo referencia esta variable, entonces vamos a decirle que es cantidad de personas, primera cosa. Cambié el nombre de la variable aquí, dio error aquí, porque obviamente esta variable ya no existe.

[02:09] Solo la reemplazo y listo. Esta es la primera mejora del código que yo estoy haciendo. Siempre recuerden declarar las variables que sean autoexplicativas. Segunda cosa es este if else dentro de este else. Ese es un anti-patrón que normalmente se lo conoce como complejidad ciclomática, que en este caso puede ser tranquilamente reducido a la primera expresión del if. ¿Por qué?

[02:42] Porque lo que yo acá estoy diciendo es si tienes más de 18 años o si la cantidad de personas es mayor o igual a 2, en cualquier caso puedes entrar. Entonces aquí yo puedo usar el operador o, siguiendo simple lógica, podría ser si la edad es mayor o igual a 18 o viene acompañado. Entonces, esto debería ser true. ¿Cómo yo expreso esta condición en código?

[03:17] Muy fácil. Tenemos dos pipes. Dos pipes sobre el condicional or. Siguiendo lógica básica, él va a evaluar. Si es true aquí, entonces entra. O si es true del otro lado, también va a entrar aquí. Entonces, decimos que si cantidad de personas, vamos a copiar todo esto, o si la cantidad de personas es igual a 2, entonces usted puede entrar. Ya no necesitamos aquí este if, entonces lo borramos.

[03:56] Y vamos a borrarlo también con este else porque no lo necesitamos, nuevamente para que el código se vea mucho mejor vamos a ver la source, vamos a formatear ese código. Y perfecto, ya tenemos un código mucho más legible, ya hemos ahorrado muchas líneas de código y ya tenemos esta expresión evaluada aquí en una sola línea. Vamos a guardar y le damos play.

[04:24] Y vemos claramente que usted puede entrar porque está cumpliendo con las dos condiciones. Ahora, aquí en la evaluación de estas condiciones, recuerden siempre, el primero va a evaluar esta condición. Si esta condición es true, es verdadera, entonces ya va a ignorar la segunda condición porque no tiene sentido evaluarla si la primera ya dio verdadero y el operador es un o.

[04:53] Ahora, ¿qué pasaría si yo quiero decirle que tiene que ser mayor de 18 años y que solamente puede entrar en pareja, por ejemplo? Si llega uno solo no puede entrar. Aparte de ser mayor de edad, solamente pueden entrar en parejas. Entonces, esta condición cambia y reemplazamos por dos operadores `unperson`, no es este de acá, disculpen, pues este de acá.

[05:19] Y aquí cambia totalmente la condición porque aquí le decimos: tiene que ser mayor de 18 años y la cantidad de personas tiene que ser mayor o igual a dos. En este caso digamos que son menores de edad. No va a cumplir una condición. Guardamos, le damos play y "usted no está permitido entrar". Perfecto. Volvemos aquí.

[05:51] Ahora, yendo aún un poco más profundo, optimizando mejor el código, ya hemos visto algunos tipos de variable para declarar strings, números grandes, números medianos, números pequeños, pero existe otro tipo de variable llamado boolean.

[06:14] Como ustedes pueden ver, boolean también es una palabra reservada de Java, también es una variable del tipo primitivo, y boolean tiene una característica muy especial y es que solamente puede almacenar dos valores: `true` o `false`. Por ejemplo, boolean condición solamente puede ser igual a `true` o `false`.

[06:42] Entonces, hay otros lenguajes de programación donde los valores booleados soportan por ejemplo 0 en caso de que sea `false`, 1 en caso de que sea `true` y él lo entiende perfectamente, pero en este caso no es así, boolean tiene que recibir necesariamente `true` o `false`. No puede ser `true`, porque esto ya es un string y no es lo que boolean espera.

[07:14] Entonces, vamos a llamar a esta condición es pareja, y aquí vamos a decirle `true`. Entonces, lo que estamos diciéndole es, vamos a cambiar la edad a 21 por ejemplo, y aquí le decimos si es mayor de edad y si es pareja. "`Ctrl + espacio`", es pareja, igual igual `true`.

[07:46] Ahora, recuerden que en Java la comparación es con igual igual, eso es para comparar, porque si nosotros dijéramos es pareja igual true, un solo signo igual significa atribución de valor. Cuando nosotros decimos que es pareja, es igual a true. Entonces le asignamos el valor true a esta variable, no es lo que queremos aquí. En Java, para evaluar si es igual, es doble signo igual.

[08:19] Y otra cosa que hay aquí es que es pareja ya es una condición en sí. Entonces, preguntar si es pareja, es igual igual a true o es igual igual a false, sería una redundancia porque él ya tiene este valor aquí asignado. Esta variable de por sí ya es true o ya es false. Entonces podemos dejarlo tan simple como esto.

[08:45] Preguntar si la edad es mayor a 18 y es pareja. Perfecto. Entonces, guardamos, ejecutamos el código y listo. Como ya tenemos 21 años y es pareja, entonces usted puede entrar. Ahora, ¿yo cómo calculo si es pareja cuando la cantidad de personas es mayor a 1?

[09:15] Y si yo solamente declaro es pareja como un true, mi código no está quedando tan autoentendible, porque yo no sé en qué me estoy basando para decir que es pareja. Entonces, recuerden que esto de aquí es una expresión booleana, esto de aquí es una expresión booleana. ¿Qué es una expresión booleana? Es una expresión de código que como resultado me da true o me da false. Es una operación que me da true o false.

[09:42] Entonces, yo puedo decir tranquilamente aquí que es pareja es el resultado de cantidad de personas mayor a 1, por ejemplo. Cantidad de personas mayor a 1 va a retornar true o va a retornar false. Ese valor va a ser asignado a la variable es pareja. De esta forma yo ya tengo aquí mi código un poco más entendible. Vamos a guardar, vamos a ejecutar. Perfecto, usted puede entrar.

[10:14] Y si aún así queremos saber cuál fue el resultado de esta operación, podemos usar sysout. Vamos a decirle: El valor de la condición es, dos puntos,

nuestro operador de concatenación y nuestra variable booleana es pareja. Perfecto. Él aquí va a imprimir el resultado de esta expresión booleana. Guardamos aquí, le damos play, y el valor de la condición es true.

[10:53] Recuerden, a diferencia de otros lenguajes, en Java tiene que estar asignado un valor true o un valor false. Por ejemplo, en otros lenguajes como JavaScript, una expresión booleana podría ser siempre cantidad de personas. ¿Por qué? Porque valida que cantidad de personas no sea undefined o no sea nule. En este caso en Java no es así.

[11:17] En Java, si nosotros queremos evaluar una condición, tiene que ser un booleano. Por ejemplo, si nosotros queremos aún optimizar mucho más el código, a modo de práctica podemos hacer algo simple como cortar el código de aquí, definimos otra variable booleana preguntando: ¿puede entrar? Y eso va a ser igual a la expresión booleana que estaba aquí dentro.

[11:47] Por lo tanto nosotros aquí preguntamos directamente ¿puede entrar? Entonces el código queda: si, puede entrar. Entonces sería: "sea bienvenido". Y si no, "usted no está permitido a entrar".